Machine Learning Lab Week 12

Name: Chandan Chatragadda – PES2UG23CS141
Sem 5 – C
Date: 31th Oct 2025
UE23CS352A: Machine Learning Lab
Week 12: Naive Bayes Classifier

INTRODUCTION:

In this lab, we do text classification using the Naive Bayes algorithm on a subset of the PubMed 200k RCT dataset, aiming to predict abstract section labels (BACKGROUND, OBJECTIVE, METHODS, RESULTS, and CONCLUSIONS).

Part A implements the Multinomial Naive Bayes (MNB) classifier from scratch, computing class priors, likelihoods with Laplace smoothing, and evaluating predictions using count-based features.

Part B applies scikit-learn's MultinomialNB with TF-IDF features and uses GridSearchCV to tune key hyperparameters such as alpha and n-gram range for better performance.

Part C approximates the Bayes Optimal Classifier (BOC) by combining diverse models, Naive Bayes, Logistic Regression, Random Forest, Decision Tree and KNN through a Soft Voting Classifier with weighted averaging.

METHODOLOGY:

Part A: Frequency based Naive Bayes Classifier (From Scratch)

- Implemented MNB manually in the NaiveBayesClassifier class.
- Used CountVectorizer to convert text into token frequency counts.
- Calculated class priors and feature log-likelihoods (with Laplace smoothing) for prediction.

Part B: TF-IDF score based Classifier (Using scikit-learn)

- Used a Pipeline with TF-IDF-Vectorizer and scikit-learn's MultinomialNB.
- Trained an initial model and evaluated its performance.
- Performed hyperparameter tuning using GridSearchCV on the development set to find optimal tfidf and nb parameters.

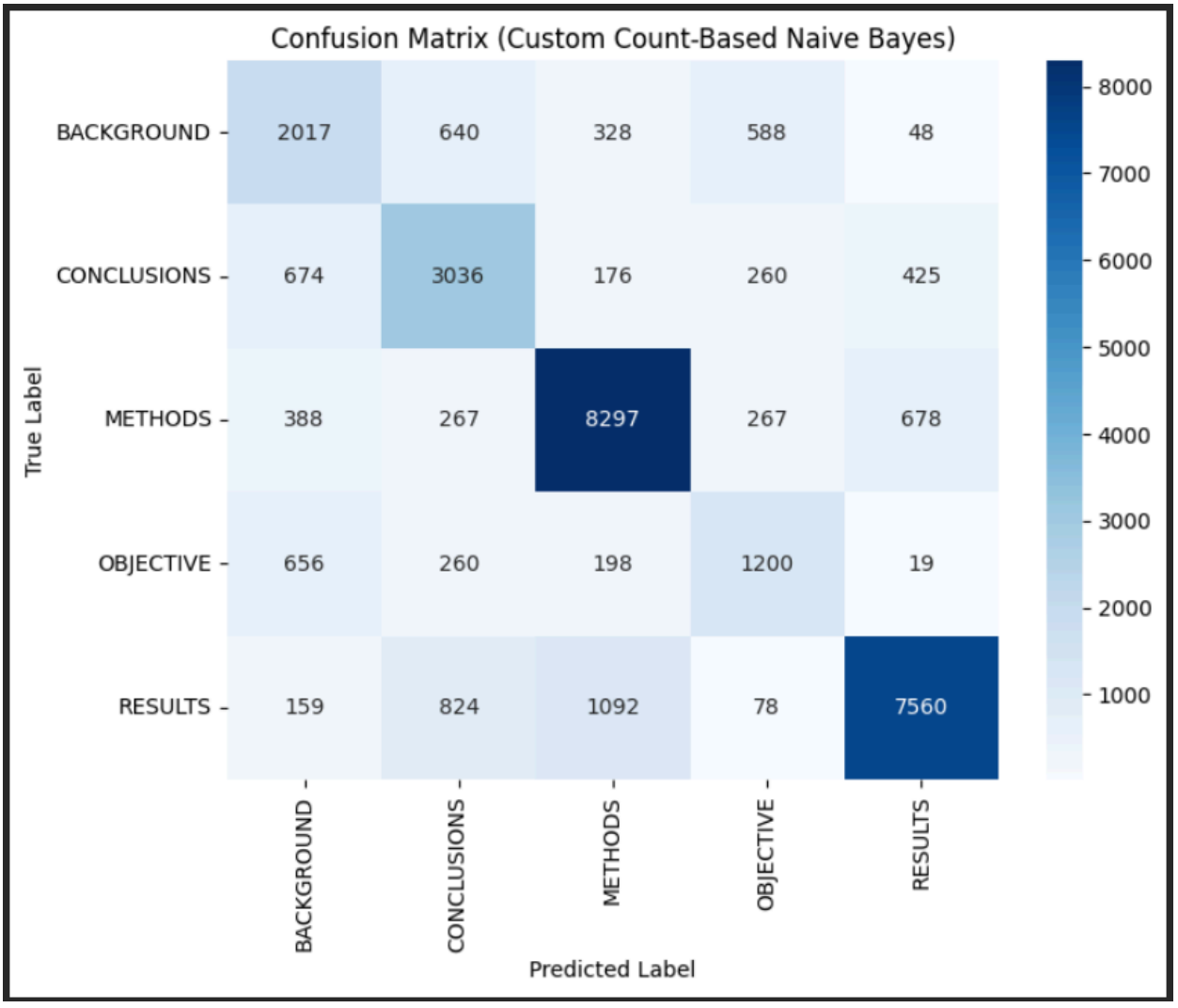
Part C: Bayes Optimal Classifier (Approximation using Soft Voting)

- Trained five diverse base models on a sampled training set.
- Calculated approximate posterior weights for each model based on their performance (log-likelihood) on a validation split of the sampled data.
- Combined the predictions of the base models using a VotingClassifier with soft voting weighted by the calculated posterior weights.

RESULTS AND ANALYSIS:

Part A:

```
=== Test Set Evaluation (Custom Count-Based Naive Bayes) ===
Accuracy: 0.7337
              precision    recall  f1-score   support

  BACKGROUND       0.52      0.56      0.54      3621
 CONCLUSIONS       0.60      0.66      0.63      4571
     METHODS       0.82      0.84      0.83      9897
   OBJECTIVE       0.50      0.51      0.51      2333
     RESULTS       0.87      0.78      0.82      9713

    accuracy                           0.73     30135
   macro avg       0.66      0.67      0.67     30135
weighted avg       0.74      0.73      0.74     30135

Macro-averaged F1 score: 0.6655
```

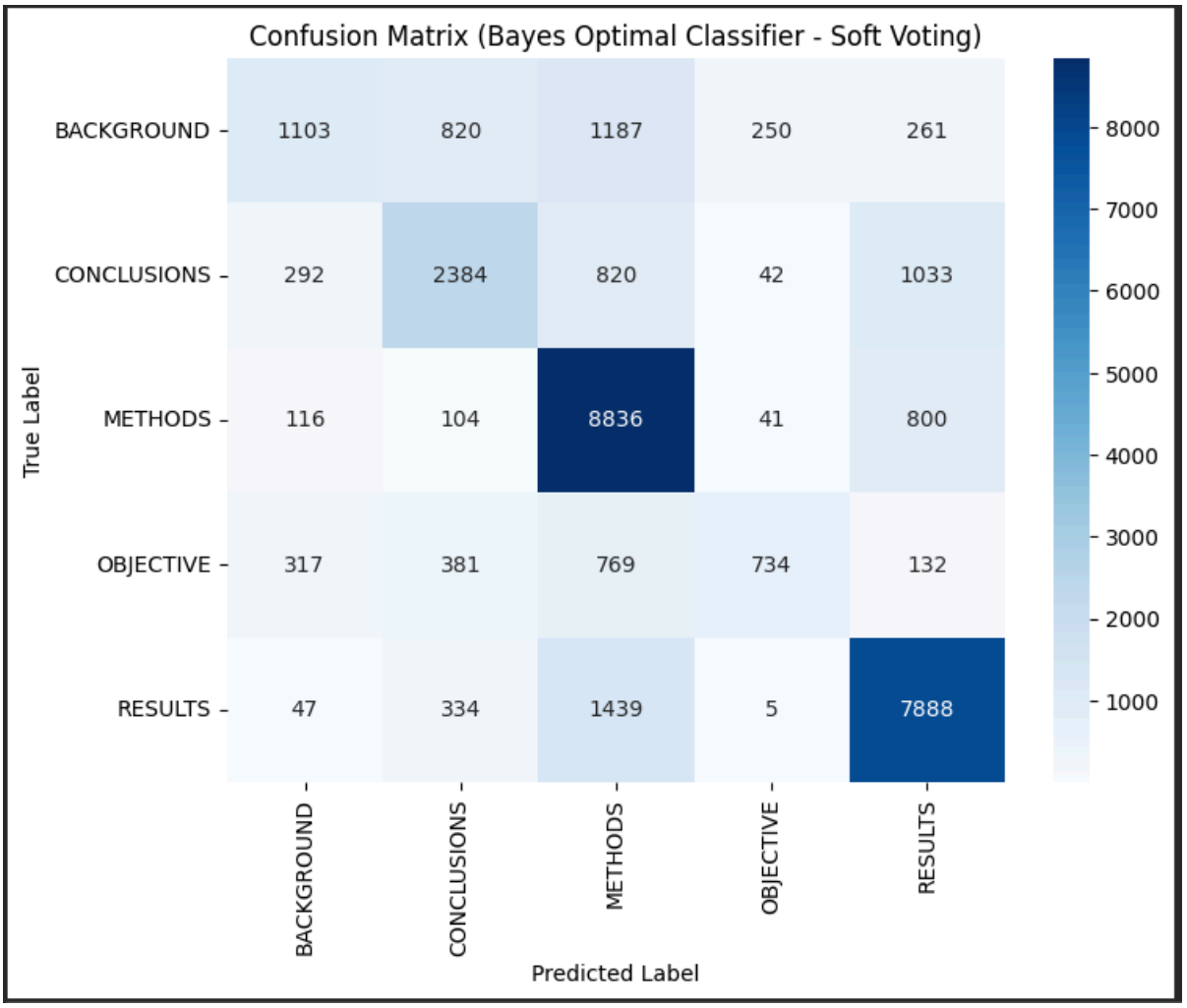


Part B:

```
Training initial Naive Bayes pipeline...
Training complete.

=== Test Set Evaluation (Initial Sklearn Model) ===
Accuracy: 0.7266
              precision    recall  f1-score   support

  BACKGROUND       0.64      0.43      0.51      3621
 CONCLUSIONS       0.62      0.61      0.62      4571
     METHODS       0.72      0.90      0.80      9897
   OBJECTIVE       0.73      0.10      0.18      2333
     RESULTS       0.80      0.87      0.83      9713

    accuracy                           0.73     30135
   macro avg       0.70      0.58      0.59     30135
weighted avg       0.72      0.73      0.70     30135

Macro-averaged F1 score: 0.5877

Starting Hyperparameter Tuning on Development Set...
Grid search complete.
Best parameters: {'nb__alpha': 0.1, 'tfidf__min_df': 10, 'tfidf__ngram_range': (1, 2)}
Best cross-validation score (macro F1): 0.6994
```

Part C:

```
Please enter your full SRN (e.g., PES1UG22CS345): PES2UG23CS141
Using dynamic sample size: 10141
Actual sampled training set size used: 10141

Training all base models...
Training NaiveBayes...
NaiveBayes training complete.
Training LogisticRegression...
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_logistic.py:1247: FutureWarni
  warnings.warn(
LogisticRegression training complete.
Training RandomForest...
RandomForest training complete.
Training DecisionTree...
DecisionTree training complete.
Training KNN...
KNN training complete.
All base models trained.

Calculating Posterior Weights...
Calculated Posterior Weights: [0.23054577 0.2500959  0.22345605 0.15514902 0.14075325]

Fitting the VotingClassifier (BOC approximation)...
Fitting complete.

Predicting on test set...

=== Final Evaluation: Bayes Optimal Classifier (Soft Voting) ===
Accuracy: 0.6950
              precision    recall  f1-score   support

  BACKGROUND       0.59      0.30      0.40      3621
 CONCLUSIONS       0.59      0.52      0.55      4571
     METHODS       0.68      0.89      0.77      9897
   OBJECTIVE       0.68      0.31      0.43      2333
     RESULTS       0.78      0.81      0.80      9713

    accuracy                           0.70     30135
   macro avg       0.66      0.57      0.59     30135
weighted avg       0.69      0.70      0.68     30135

Macro-averaged F1 score: 0.5906
```



DISCUSSION:

- Part A: Custom Count-Based Naive Bayes
  - Accuracy: 0.7337
  - Macro-averaged F1 score: 0.6655

- Part B: Tuned Sklearn TF-IDF based Naive Bayes
  - Initial Sklearn Model:
  - Accuracy: 0.7266
  - Macro-averaged F1 score: 0.5877
  - Best Parameters from Grid Search (on Dev Set): {'nb__alpha': 0.1, 'tfidf__min_df': 10, 'tfidf__ngram_range': (1, 2)}
  - Performance with Best Parameters: From the 0.6994 Macro F1, we can infer that the tuned model performs better than the initial Sklearn model on unseen data.

- Part C: Bayes Optimal Classifier (Soft Voting Approximation)
  - Accuracy: 0.6950
  - Macro-averaged F1 score: 0.5906

Comparison:

- The Custom Count-Based Naive Bayes achieved the highest Accuracy and Macro-averaged F1 score.
- The Initial Sklearn TF-IDF based Naive Bayes had slightly lower performance than the custom model.
- The Bayes Optimal Classifier approximation showed slightly lower performance than the custom MNB model.